TITLE: Method and apparatus for label switching data packets

FIELD OF THE INVENTION

The invention relates to the transport of data packets
5   in a data network, more specifically to a method and an
apparatus for label switching data packets. The invention
also provides a novel data packet containing information
allowing to ascertain if one or more labels in the data
packet have switching significance in a certain layer of the
10   data network.

BACKGROUND OF THE INVENTION

Multiprotocol Label Switching (MPLS) is a forwarding
protocol, which uses labels to increase the speed and
15   efficiency with which a data network transports data
packets. In contrast to a traditional IP address, the MPLS
protocol uses a limited set of labels on the basis of which
switching decisions are made into an MPLS domain. Each data
packet contains a label stack that can hold one or more
20   labels.

When a data packet enters an MPLS domain via a Label
Edge Router (LER), the LER determines if the data packet
should be accepted and in the affirmative pushes one or more
25   labels in the label stack. The labels determine the path or
part of the path the data packet should follow in the MPLS
domain. Next, the data packet leaves the LER and reaches a
Label Switching Router (LSR) that examines the label at the
topmost position in the label stack and determines how the
30   data packet should be switched. The LSR can perform label
stack manipulations, such as swapping or pushing other
labels in the label stack that will determine where the data
packet will go next. When the data packet reaches the end

of the MPLS domain, it can egress the domain via another LER, which pops all the labels in the stack thus restoring the data packet to its condition prior to entering the MPLS domain.

5

Since different parts of a large data network are usually managed by different entities, such as different carriers, there is generally no network-wide label convention among the carriers. Accordingly, the same label can have a different switching significance in different locations of the network. The use of a stack for temporarily storing labels in the data packet ensures that only the correct label is presented for processing at the location where it has a proper switching significance. This however holds true only if all the stack manipulations occurring along the path of the data packet are flawless. If for some reason an error is made and a label is incorrectly pushed in the stack or popped from the stack, the remaining labels will be incorrectly interpreted at downstream LSRs with the result that the data packet will follow an erroneous path. A similar problem may occur if a connection error exists between two LSRs in the data network. The connection error will cause data packets to be incorrectly received by an LSR that however, will not be able to recognize the problem and will simply incorrectly forward the data packets at other nodes.

Another drawback associated with currently implemented MPLS protocols is their inability to provide a clear demarcation between the network and its users. Accordingly, it is difficult for the data network to enforce admission control, hence compliance with a Service Level Agreement (SLA) or guarantee network performance

such as Quality of Service (QoS) arrangements.

    Accordingly, there exists a need in the industry to
provide a novel mechanism for data packet transmission in
5 a data network, based on label switching that avoids at
least some of the problems associated with prior art
methods.

SUMMARY OF THE INVENTION
    In a first broad aspect, the invention provides a
10 data packet for transmission in a data network having one
or more layers.   The data packet has a label stack with
one or more labels on the basis of which the data packet
can be switched in the data network, where each label has
a switching significance in one or more of the layers of
15 the data network.   The data packet also has at least one
layer identifier associated with a label in the label
stack.

    Advantageously, this data packet structure allows
network entities in the data network that receive the data
20 packet to determine where the labels in the data packet
have switching significance.   Accordingly, misdirected
data packets can be identified at an early stage and
prevented from further propagating along an erroneous
path.   In addition, the layer identifiers permit a clearer
25 demarcation between the network and its users, allowing
network operators to better enforce compliance with SLAs
or to meet QoS agreements.

    In a second broad aspect, the invention provides a
30 network entity for use in a data network for label
switching data packets, the data network having one or
more layers. The network entity has one or more input
ports for receiving data packets to be label switched and

one or more output ports for releasing data packets from
the network entity.   The network entity also has a
switching controller for switching a data packet received
at one of the input ports to one of said output ports on a
5   basis of a label contained in the data packet, the
switching controller operative to ascertain if the label
contained in the data packet has a switching significance
in the layer of the data network to which the network
entity belongs.

10

     In a third broad aspect the invention provides a
method for label switching data packets. The method
comprises receiving a data packet containing at least one
label. The method further includes ascertaining if the
15   label has a switching significance in an area of interest
in the data network and in the affirmative, switching the
data packet on the basis of the label.

20

BRIEF DESCRIPTION OF THE DRAWINGS
     A detailed description of examples of implementation
of the present invention is provided hereinbelow with
reference to the following drawings, in which:

25

     Figure 1 is a diagram illustrating the structure of a
data packet in accordance with a non-limiting example of
the invention;

30   Figure 2 is a schematical diagram of a label-switched
network domain in which the data packet shown at Figure 1
can be transported; and

     Figure 3 is a block diagram of a network entity used

in the network shown at Figure 2.

      In the drawings, embodiments of the invention are
illustrated by way of example.   It is to be expressly
5 understood that the description and drawings are only for
purposes of illustration and as an aid to understanding,
and are not intended to be a definition of the limits of
the invention.

10 **DETAILED DESCRIPTION**
      Figure 1 illustrates an example of implementation of
a data packet in accordance with the present invention.
This data packet structure can be used to encapsulate
packetized traffic for transport over a label-switched
15 transport network. In a specific example, the data packet
can be generated by using the General Framing Procedure
(GFP), described in the ITU-T recommendation G.7041, which
defines a generic framing procedure to delineate variable
length payloads to higher-level signal clients for
20 subsequent transport into payload envelopes.

      It should be expressly noted that the use of the GFP
packet delineation scheme is only an example of
implementation of the invention and other schemes to
25 obtain the data packet can be used without departing from
the spirit of the invention.

      The data packet of Figure 1 is label switched and
comprises an overhead component having a label container
30 10 that includes two sections namely, a label stack 12 and
a label stack management field 14. The label stack
management field 14 is a 4-byte field that includes
information relative to the label container 10 and other

information of a more general nature relative to the data packet. For example, the label stack management field 14 includes the size of the label stack 12, the Class of Service (CoS) associated with the data packet, the data 5 packet loss priority (discard eligibility), the label container error check, among other possible types of information.

The label stack 12 includes one or more entries 16. 10 Each entry 16 is 32-bit wide and has a 20-bit label field 18 that contains a label. The entry 16 further includes a 4-bit field 20 holding a layer identifier (LID) associated with the label in the label field 18. The layer identifier indicates in which network layer the label has 15 a switching significance. The notion of layer in a data network and the notion of switching significance will be discussed later.

The entry 16 further includes a field 22, holding a 20 local or foreign (L/F) key capable to acquire different values. The key is an additional hierarchical level, above the layer identifier, permitting to distinguish between network layers having identical layer identifiers. In a specific example of implementation, the L/F key is a 25 one-bit flag that can take two values. The entry 16 also includes a 2-bit Operations and Maintenance (OAM) field 24 for holding an OAM alert indication, a 1-bit Forward Congestion Notification (FECN) field 26, a 1-bit Backward Congestion Notification (BECN) field 28, and an additional 30 3-bit field 30 for supplementary information, if required by the intended application.

As it will be described later, the size of the label stack 12 will vary from one data packet to another, as a

result of entries 16 being pushed into the label stack 12
and entries 16 being popped from the label stack 12.

   In addition to the label container 10 described
5 above, the data packet also has a payload in which is
placed the data being transported.   The payload is not
shown   in   Figure   1.   In   a   specific   example   of
implementation, the payload may be an Internet Protocol
(IP) packet that includes an address designating the
10 destination of the packet.   If desired, the data packet
can include other overhead components such as error
detection and/or error correction fields, among others.

   Figure 2 is a schematical diagram of a data network
15 in which the data packet described earlier can be used.
An Ethernet client 32 is connected to the edge of the data
network.   The data flow is received by the network entity
34, which may be a router, and which encapsulates the
packets from the client 32, by using the GFP scheme into
20 data packets shown in Figure 1.   Those data packets are
delivered via a sub-network connection 36 to a network
entity 38, such as an LSR that will multiplex those data
packets with additional traffic.   At this point, the
network entity 38 will push an entry 16 in the label stack
25 12. The entry 16 includes a label and a layer identifier.
The purpose of the layer identifier is to convey in which
layer of the data network the label has switching
significance.   A label is considered to have a switching
significance when the switching behavior produced on the
30 basis of the label will be predictable and as intended.

   The notion of a layer in a data network is generally
known and does not need to be described in detail.
Suffice it to say that a layer encompasses one or more

network entities that share a common control plane, which
can be implemented as a distributed entity or a
centralized one. A layer may encompass a single network
entity or a plurality of entities. In the case of a
plurality of entities, those entities can be contiguous,
in other words directly connected to one another via
links. Alternatively, the network entities may not be
directly connected to one another. Instead they may be
connected via entities or links that belong to a different
layer.

Layers in a data network are normally attributed in
an arbitrary manner that may take into account topological
considerations and owner/operator considerations.

A general block diagram of the network entity 38 is
shown at Figure 3. The network entity 38 is in the form
of a router, comprising a plurality of input ports 40 at
which data packets are received and a plurality of output
ports 42 from which data packets are forwarded to
respective destinations. The network entity 38 also
includes a switching controller 44 that switches the data
packets received at the input ports 40 to their respective
output ports 42. The switching controller 44 performs
switching operations on the basis of the labels contained
in the label stacks 12 of the data packets. Specifically,
when a data packet is received at an input port 40, it is
loaded in the computer readable storage medium of the
switching controller 44. The entry 16 in the topmost
position in the label stack 12 is read. The switching
controller 44 will extract the layer identifier and
compare it to the layer identifier to which the network
entity 38 belongs. If there is a match, the switching
controller 44 concludes that the label in the topmost

entry 16 in the label stack 12 has a switching significance in the network layer to which the network entity 38 belongs, and the switching controller 44 will switch the data packet on the basis of the label.

5 However, if there is no match, the switching controller 44 will declare an error condition and will not label switch the data packet. The error condition may cause the network entity 38 to generate one or more error notifications to other entities in the network, such that

10 they are made aware of the problem. Other ways of handling the error condition can be envisaged without departing from the spirit of the invention.

The switching controller 44 can also perform

15 manipulations of the label stack 12 such as pushing additional labels and associated layer identifiers (i.e. multiplexing the stream onto a server layer), or popping labels and associated layer identifiers (i.e. demultiplexing from a server layer), among others.

20

The switching controller 44 can be implemented in software, hardware or a combination of both.

Referring back to the network diagram of Figure 2 and

25 continuing with the previous example, the data network has two layers, the first layer including network entities 38 and 66, and a second layer including network entities 46, 50, 52, 54, 56 and 58. When the data packet is released from the network entity 38 it is forwarded to the network

30 entity 46 via sub network connection 48. The data packet travels toward the network entity 46 embedded into a signal transmitted over a suitable link, which can be wireline or wireless link. At network entity 46 a second label and the associated layer identifier are pushed into

the label stack 12. The data packet is forwarded through
the second layer successively via the network entities 50,
52, 54, 56 and 58 and the sub network connections 60, 62,
and 64. At each network entity, before performing any
5 label switching, the layer identifier is assessed and the
network entity determines if the label has switching
significance. In the affirmative, the switching operation
is performed; otherwise an error condition is declared.

10   The data packet leaves the second layer and enters
back the first layer when it is forwarded from the network
entity 58 to the network entity 66 via the sub network
connection 64. The same layer verification is performed
here, as the one described earlier. The topmost entry 16
15 in the label stack 12 is read and the layer identifier is
extracted to determine if the label has a switching
significance in the second layer of the data network. In
the affirmative, the label switching operation is
performed, otherwise an error condition is declared. For
20 instance, assume that as a result of an error, the data
packet is switched from the network entity 56 directly to
the network entity 66 (along the dotted line connection),
instead of being forwarded to the network entity 58. The
network entity 66 checks the layer identifier in the
25 topmost entry 16 in the label stack 12 and will declare an
error condition, thus preventing further propagation of
the data packet along an erroneous path.

    Since no further label switching will be performed
30 downstream the network entity 66, the latter will clear
the label stack 12. The resulting data packet is sent to
the edge network entity 68 where the data packet undergoes
a de-encapsulation process to obtain the original data
packet that was forwarded by the client 32.

In the above example, an assumption was made to the
effect that a convention exists in the data network as to
the layer identifiers.   Specifically, the convention
5 assigns a unique identifier to each layer such that there
is no confusion.   This scenario may be realistic for a
portion of a data network managed by a single entity, such
as a carrier. However, in large networks, where different
parts may be under the control of different carriers, such
10 convention may not exist, which may result into conflicts
if the same layer identifier is assigned to more than one
layer.   To avoid this potential problem, the data packet
provides in each entry 16 an L/F key that can be used to
indicate that a certain layer identifier is foreign or
15 local.   By default, this key, which in this example is
implemented as a one-bit flag, is set to local.   However,
when a first layer transits the network of a second
administration using a second layer, the flag is set to
foreign by the network entity at the boundary between the
20 two administrations.   In this fashion, should a traffic
spill occur and a network entity is presented with a label
having a "foreign" layer identifier, an error condition is
declared.

25   To better illustrate this feature consider a
situation where the network entity is located at a logical
boundary between a first network operator and a second
network operator.   The second network operator can
terminate a layer of the data network implemented or
30 managed by the first network operator by overwriting the
key in the data packet associated with the identifier of
the first layer with a "local" value that corresponds to a
key of a layer implemented or managed by the second
network operator.

In another situation where the network entity is also located at the logical boundary between the first network operator and the second network operator, the network
5 entity is capable of overwriting a key in a data packet associated with a layer identifier of a layer managed by the first network operator with a "foreign" value indicating that the layer identifier is not valid within a network layer implemented or managed by the second network
10 operator. The foreign flag is set when the second network operator encapsulates the data from the first network operator, i.e., when a new label is pushed onto the label stack.

15 Although various embodiments have been illustrated, this was for the purpose of describing, but not limiting, the invention. Various modifications will become apparent to those skilled in the art and are within the scope of this invention, which is defined more particularly by the
20 attached claims.